Analyse des transitoires par masques de Gabor

grégori Robinson

le 29 juin 2007

Table des matières

Introduction **3**

I Modèles simples de transition 4

1 Introduction **5**
1.1 Déroulement 5
1.2 Intérêt de la création de formes simples d'attaque 5

2 Choix et création des formes de transition **6**
2.1 Choix des formes d'enveloppe 6
2.2 Paramètres à prendre en compte 7
2.2.1 Choix des paramètres 7
2.2.2 Nouvelles formules 8
2.3 Mode d'emploi des fonctions de création de formes d'attaque . . 9
2.3.1 Fonction genere_trans.m 9
2.3.2 Script recup_donnee.m 10

II Analyse des transitoires en Gabor 12

3 Théorie des masques de Gabor **13**
3.1 Bases sur l'utilisation des transformées temps-fréquence en analyse synthèse 13
3.2 La transformée de Gabor 14
3.2.1 Rappels 14
3.2.2 Toolbox utilisée - tests 14
3.3 Les multiplicateurs de Gabor 19
3.3.1 Définition 19
3.3.2 Utilisation 19

4 Optimisation des masques de Gabor pour caractériser les transitoires **21**
4.1 Intérêt pour caractériser les transitoires 21
4.2 Calcul du masque sans recalage 21

4.3 Calcul du masque avec recalage 22
4.4 Les différentes méthodes de recalage 25
4.4.1 Recalage par centre de gravité 25
4.4.2 recalage par corrélation 25
4.5 Algorithme itératif 27

Conclusion **30**

Introduction

Au cours de mon année de Master 2 en signal et trajectographie à l'UFR des sciences et techniques de l'université du sud Toulon-Var, jai été amené a effectuer un stage de quatre mois au Laboratoire de Mécanique et d'Accoustique (LMA) du CNRS (Centre National de la Recherche Scientifique) de Marseille. Le sujet de mon travail s'intitule : Etude et caractérisation des transitoires d'attaque par des méthodes temps-fréquence.
Monsieur Richard Kronland du LMA et Monsieur Philippe Depalle de l'université de McGill m'ont encadrés pendant toute la durée du stage qui a été réalisé en collaboration avec Monsieur Bruno Torresani du LATP de Marseille.

Pour caractériser un son on peut le décomposer en trois parties additives : la partie tonale (où sinusoïdale), la partie transitoire localisée en temps et étalée en fréquence, et la partie stochastique stationnaire [2].
La partie transitoire contient toutes les variations rapides du signal. Il est capital de savoir la modéliser correctement en audio car elle peut influencer la perception que l'on a d'un son. Cela nous permettra par exemple de caractériser la forme d'attaque d'une note jouée par un instrument de musique spécifique (en effet chaque type d'instrument possède une attaque qui lui est propre) mais aussi de détecter l'instant de transition afin de pouvoir segmenter un morceaux où en détecter le rythme ou le tempo.
Nous avons choisi de travailler en temps fréquence à cause de la nature des transitoires. En effet, il existe une multitude de formes d'attaques possibles et en temporel comme en fréquentiel, des formes portant la même information peuvent ne pas se ressembler. De plus, nous avons opté pour les multiplicateurs de Gabor car cette technique ayant eu des développements théoriques récents permet de travailler au maximum sur le signal et non pas sur des algorythmes informatiques, et de valider les résultats obtenus par synthèse. De plus, cette méthode peut être appliquée de manière plus générale à tous les types de sons.
Nous avons donc dans un premier temps cherché à générer des formes de transition simples permettant de tester les méthodes d'analyse puis nous nous sommes penchés sur la problématique des multiplicateurs de Gabor et de leur utilisation dans la caractérisation des transitoires.

Première partie

Modèles simples de transition

Chapitre 1

Introduction

1.1 Déroulement

Lors de la première partie de mon stage il a fallu d'abord choisir quelles formes typiques de transition représenter puis mettre en place des routines sous *Matlab*© afin de générer des signaux composites où chaque composante possède sa propore enveloppe temporelle. Cela consiste à générer des formes simples d'attaque, c'est à dire de transition entre deux états stationnaires (Ces formes seront aussi appelées transitoires dans la suite du document) au moyen d'enveloppes appliquées ensuite à des signaux quelconques.

1.2 Intérêt de la création de formes simples d'attaque

Le but est de trouver tout d'abord des formes d'attaque (enveloppe de la sinusoïde) permettant de catégoriser les différents transitoires. En effet, il faut, au vue des transitoires habituellement dans les sons réels, sélectionner les formes simples leur correspondant le mieux afin d'obtenir des signaux calibrés. Ces signaux calibrés serviront à évaluer la méthode d'analyse des transitoires mise en place, car ils permettent de voir si ses résultats sont cohérents avant d'effectuer des tests sur des signaux réels. Ainsi les résultats sur des signaux réels plus complexes peuvent aussi être pressentis à partir des résultats obtenus lors de ces tests.

Chapitre 2

Choix et création des formes de transition

2.1 Choix des formes d'enveloppe

Les formes d'enveloppe de l'attaque ont été choisies afin de représenter au mieux les différents transitoires. En effet, on travaille dans le domaine temps-fréquence et donc l'étalement spectral est un paramètre important. Cet étalement du spectre des transitoires dans les sons dépend du nombre de dérivées continues que possède la forme d'enveloppe et donc de la classe de la fonction qui la décrit. Plus la classe de la fonction d'enveloppe est élevée et plus la pente du spectre va diminuer en valeur absolue : on aura ainsi un spectre plus étalé pour des transitions de classe petite que pour des transitions de classe élevée. Pour bien représenter les principales possibilités il a fallu choisir des enveloppes de classe C_0 (pente à 6 dB/octave en puissance), C_1 (pente à 12 dB/octave en puissance) et C_2 (pente à 18dB/octave en puissance) et trouver des expressions mathématiques les caractérisant.
Sachant que ces expressions ne sont valable que dans l'intervalle de durée de l'enveloppe, c'est à dire que l'on part de t=R (où elle a une valeur nulle) et qu'elle redevient constante lorsque sa valeur arrive à 1, voici les formules des différentes enveloppes sélectionnées :
Pour la classe C_0 la fonction rectangulaire a bien entendu été choisie (cf. figure 2.1).
Les fonctions de classe C_1 ont été représentées par des attaques linéaire et exponentielle : les formules mathématiques de ces enveloppes sont :

- Bien évidemment $S(t) = at + b$ pour l'attaque linéaire
- et $S(t) = 1 - e^{\frac{-\alpha t}{t_0}}$ pour l'attaque exponentielle

Pour les fonctions de classe C_2, on a choisi :

- une enveloppe parabolique (polynôme de degré 2) avec $S(t) = at^2 + bt + c$
- une enveloppe en forme de cosinus soit $S(t) = \frac{1-\cos \omega t}{2}$

On peut voir des exemples des ces enveloppes dans la figure 2.1

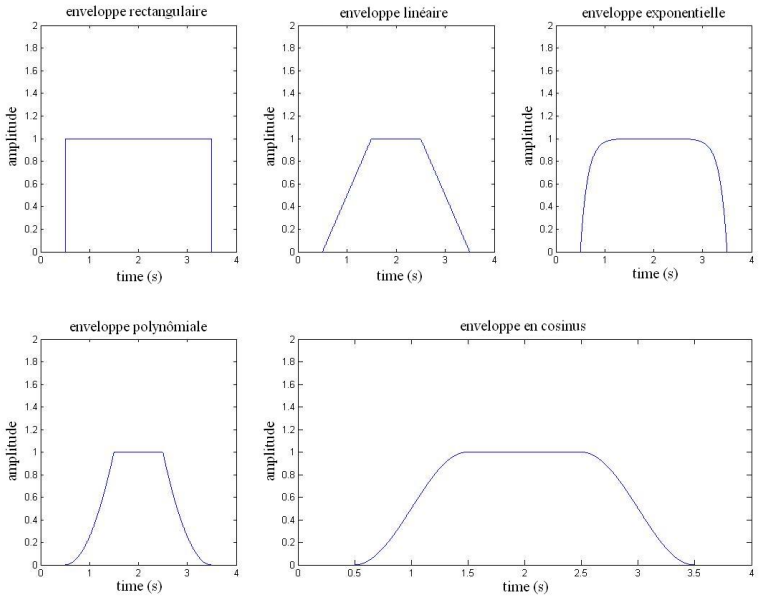

FIG. 2.1 – Exemples d'enveloppes avec un retard $R = 0,5$ s s'éteignant de façon symétrique par rapport à leur naissance.

Le transitoire sera obtenu en multipliant ces enveloppes à un signal quelconque. Pour les différents tests, elles ont été appliquées à des exponentielles complexes à fréquences données car cela nous permet d'obtenir des signaux calibrés plus faciles à interpréter.

2.2 Paramètres à prendre en compte

2.2.1 Choix des paramètres

Les différentes fonctions données ci-dessus doivent toutes êtres rêglées par des paramètres afin de modifier leur forme. Nous avons choisi de ne pas garder les paramètres de la formule mathématique mais de nous intéresser à d'autres paramètres ayant un sens plus perceptif. Les formes d'attaque seront donc exprimées en fonction :

1. Pour l'enveloppe :
   - du temps que met l'enveloppe pour passer de 0 à 1 (cette variable sera appelée temps d'attaque et notée τ dans la suite du rapport)
   - du retard du signal par rapport à l'origine des temps (noté R) c'est à dire l'instant (positif) où l'enveloppe commence à croître.
2. Pour la partie stationnaire

- de la fréquence de l'exponentielle complexe (notée ensuite ν)
- de la phase initiale de cette exponentielle (ϕ)

2.2.2 Nouvelles formules

Pour chaque enveloppe, les formules ont été exprimées en fonction du temps d'attaque τ, les expressions suivantes n'étant toujours valables que de $t = R$ à $t = R + \tau$:

- l'enveloppe rectangulaire a un temps d'attaque nul
- pour l'enveloppe linéaire on a :

$$S(t) = \frac{1}{\tau}(t - R) \tag{2.1}$$

- le problème de l'enveloppe exponentielle est qu'elle n'arrive jamais à 1. On est alors obligé de faire une approximation. La précision étant fixée à 10^{-3} on obtient alors[1] :

$$1 - e^{\frac{-\tau}{t_0}} = 1 - 10^{-3}$$

$$\Rightarrow t_0 = \frac{\tau}{\ln 10^3}$$

$$\Rightarrow S(t) = e^{\frac{ln10^3(t-R)}{\tau}} \tag{2.2}$$

- l'enveloppe parabolique nécéssite un autre paramètre controlant sa courbure : $c = \frac{d^2S(t)}{dt^2}$
  on a alors :

$$S(t) = \frac{c}{2}(t - R)^2 + (\frac{1}{\tau} - \frac{c\tau}{2})(t - R) \tag{2.3}$$

[1] exceptionellement l'enveloppe sera donc prolongée à la valeur 0,999 pour $t > \tau$ afin de ne pas créer de discontinuité

– pour l'enveloppe en cosinus on passe de 0 à 1 en une demi-période soit $\frac{1}{2f}$ si f est la période du cosinus. Donc la fréquence du cosinus doit être $f = \frac{1}{2\tau}$. La formule devient alors :

$$S(t) = \frac{1 - \cos\left(2\pi \frac{1}{2\tau}(t - R)\right)}{2} \tag{2.4}$$

Pour obtenir la forme de transitoire voulue il faudra ensuite multiplier ces enveloppes par $e^{i(2\pi\nu t + \phi)}$. On obtiendra alors une forme d'attaque normée à 1.

Il faut aussi choisir la façon dont le signal s'éteint. Dans le programme nous avons choisi de proposer trois choix :

1. symétriquement par rapport à sa naissance
2. en exponentielle décroissante : $e^{-\alpha t}$
3. ne pas faire s'éteindre le signal (il reste à 1 jusqu'à la fin)

2.3 Mode d'emploi des fonctions de création de formes d'attaque

Pour créer des formes d'attaque sous $Matlab^{©}$, une toolbox a été développée lors du stage. On peut en retenir les deux fonctions principales :

– la fonction genere_trans.m qui génere le signal s(t) correspondant à la forme d'attaque voulue [2]
– le script recup_donnee.m permettant de recupérer des données dans un fichier texte afin de pouvoir les utiliser avec genere_trans.m

2.3.1 Fonction genere_trans.m

La fonction genere_trans.m se présente sous cette forme :
– $[s, t] = genere\_trans(tau, W, F, A, T, D, Phi, M)$.

Toutes les entrées sont des vecteurs ayant pour longueur le nombre de sinusoïdes que l'on veut sommer. Dans l'ordre, les entrées sont :
– tau, vecteur contenant le temps d'attaque τ de chaque enveloppe.

[2] d'autres fonctions sont appelées lors du lancement, il est donc necéssaire de disposer de toute la toolbox pour la faire fonctionner

- W, vecteur contenant des valeurs correspondant à la forme de chaque enveloppe :
  1. rectangulaire
  2. linéaire
  3. parabolique (la courbure c est demandée dans la fenêtre de commande après le lancement de la fonction)
  4. exponentielle
  5. cosinus
- F, vecteur des fréquences des sinusoïdes
- A, vecteur des amplitudes des sinusoïdes
- T, vecteur des retards
- D, état stationnaire c'est à dire le temps pendant lequel l'enveloppe reste à 1 après avoir terminé l'attaque
- Phi, vecteur des phases initiales ϕ
- vecteur contenant des valeurs correspondant à la façon dont meure chaque signal :
  1. symétrique
  2. exponentiel (le α est demandé dans la fenêtre de commande)
  3. pas de mort (le signal reste à 1)

Les sorties sont le signal s (dont la longueur est adaptée aux entrées que l'on a choisie), et le vecteur temps t correspondant.

2.3.2 Script recup_donnee.m

Ce script sera surtout utile pour les tests afin de récupérer les données dans un fichiers texte contenant les paramètres voulus. Il permet ainsi de ne pas avoir à écrire à chaque fois les entrées dans la fonction genere_trans.m. Pour lancer ce script il suffit de mettre en entrée le nom du fichier texte à traiter en chaine de caractères. Par exemple, si on veut traiter le fichier data.txt on tape :

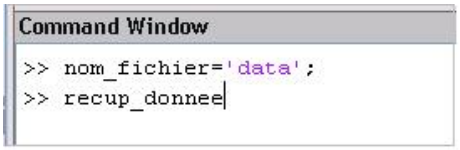

FIG. 2.2 – fenêtre de commande

Le fichier de données doit avoir un format bien précis : on en montre un exemple ci dessous où tous les types d'enveloppe et de fin de signal sont représentés.

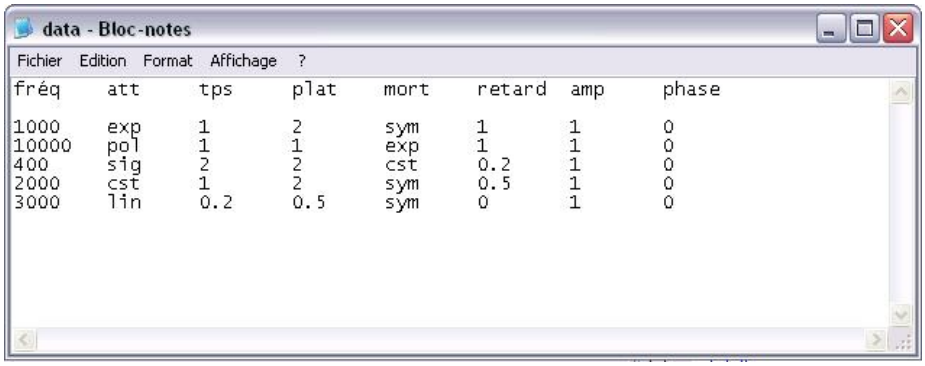
data - Bloc-notes
Fichier Edition Format Affichage ?
fréq att tps plat mort retard amp phase
1000 exp 1 2 sym 1 1 0
10000 pol 1 1 exp 1 1 0
400 sig 2 2 cst 0.2 1 0
2000 cst 1 2 sym 0.5 1 0
3000 lin 0.2 0.5 sym 0 1 0

FIG. 2.3 – fichier de données

Une fois les données récupérées il suffit d'appeler la fonction genere_trans.m pour générer les signaux voulus.

Deuxième partie

Analyse des transitoires en Gabor

Chapitre 3

Théorie des masques de Gabor

3.1 Bases sur l'utilisation des transformées temps-fréquence en analyse synthèse

Lorsqu'on a besoin de connaitre à la fois les propriétés en temps et en fréquence d'un signal, on ne peut alors plus observer ni le signal temporel, ni sa transformée de Fourrier. L'idée qui vient alors est d'observer la transformée de Fourrier à travers une fenêtre qui évolue dans le temps afin de trouver l'évolution de la fréquence au cours du temps. On aboutit alors aux transformées dans le plan temps fréquence. Cette transformée varie bien évidemment en fonction de la fenêtre choisie que l'on nommera fenêtre d'analyse.
Lorsqu'on regarde maintenant un signal discret, la transformée dépend alors non seulement de cette fenêtre d'analyse mais aussi des paramêtres de maillage du plan temps-fréquence. En effet si ces paramêtres sont trop grand alors on perdra de l'information et la transformation ne sera pas inversible, alors que si ils sont petits on aura une redondance d'information (ce qui n'est pas un problème en soi).
Dans le cas où la transformation est inversible, on arrive à reconstruire le signal en comparant sa transformée non pas avec la fenêtre d'analyse mais avec une fenêtre résultant de l'application d'un opérateur R^{-1} dépendant du maillage du plan temps-fréquence sur la précédente fenêtre (R n'est inversible que si la transformation l'est). Cette nouvelle fonction est appelée fenêtre de synthèse.
Lorsqu'on se place dans le cas où l'on n'a pas de redondance mais où la transformation est quand même inversible (le maillage de la grille est à la limite), R est alors égal à l'opérateur identité. On dit que l'on se place dans un repère strict et on a égalité entre la fenêtre de synthèse et la fenêtre d'analyse.

3.2 La transformée de Gabor

3.2.1 Rappels

Nous allons d'abord faire un petit rappel sur la transformée de Gabor d'un signal :

Définition

Soit s(t) un signal dans $L^2(R)$ et g une fenêtre d'analyse. La transformée de Gabor de s associée à la fenêtre g est :

$$v_g(s)(k,l) = \langle s, g_{k,l} \rangle = \int s(t).g(t - kb_0).e^{2i\pi l\nu_0 t}dt \tag{3.1}$$

où

$$g_{k,l}(t) = g(t - kb_0).e^{2i\pi l\nu_0 t}$$

Elle peut donc s'interpréter comme la corrélation du signal s avec la fenêtre translatée en temps et en fréquence.
Si on définit M_ν comme l'opérateur de modulation de fréquence ν et T_b l'opérateur de translation d'un temps b, on a alors :

$$v_g(s)(k,l) = \langle s, M_{l\nu_0} T_{kb_0} g \rangle$$

Reconstruction

Si on veut ensuite reconstruire le signal s grâce à sa transformée de Gabor, on utilise la transformée inverse :

$$s = \sum_{k,l} v_g(s).h_{k,l} \tag{3.2}$$

Où h est la fenêtre de synthèse, duale de g et $h_{k,l} = h(t - kb_0).e^{2i\pi l\nu_0 t}$

3.2.2 Toolbox utilisée - tests

Lors de ce stage j'ai essentiellement utilisé une toolbox temps fréquence développée par Peter L. Sondergaard au NUHAG (Numerical Harmonic Analysis Group) de l'université de Vienne : la toolbox LTFAT (Linear Time-Frequency Analysis Toolbox)[5]. Une référence bibliographique utile pour la compréhension de la toolbox est le rapport [1].L'intérêt de cette toolbox est qu'elle contient toutes les fonctions utiles de calcul des fenêtres de synthèse et d'analyse, de calcul de transformée de Gabor directe et inverse ainsi que d'autres outils permettant de travailler sur les masques de Gabor.
Afin de nous familiariser avec cette toolbox et d'en vérifier certaines propriétés précises ainsi que la définition des outils utilisés, nous avons effectué différents tests à l'aide des signaux calibrés générés dans la première partie.

Tests sur la fonction dgt.m

La fonction dgt.m est la fonction qui calcule la transformée de Gabor d'un signal s en utilisant la fenêtre d'analyse, le pas d'échantillonnage (équivalent à b_0 dans la formule donnée ci dessus : equation 3.1) ainsi que le nombre de modulations M, c'est à dire le nombre de fréquence pour lesquelles le programme doit calculer les valeurs de la transformée. En effet la transformée est calculée avec une approche en bancs de filtres et on aura donc M filtrages lors du calcul [1].

Afin de pouvoir regarder les résultats des tests, il a fallu coder une fonction (plot_ttf.m) qui prend en entrée le vecteur des temps, celui des fréquences et une transformée calculée par dgt.m, et qui trace le module de cette transformée dans le plan temps fréquence.

Ensuite nous avons tout d'abord testé le résultat obtenu avec une impulsion située à t_0 quelconque : $s(t) = \delta_{t_0}(t)$ c'est à dire s(t)=0 si $t \neq t_0$ et $s(t_0) = 1$.

On obtient avec ce signal une transformée de Gabor tracée dans la figure 3.1 avec $t_0 = 0,5$ s. On remarque que la transformée est très localisée dans le temps et étalée en fréquence. Cela correspond à ce que l'on attendait pour une impulsion. De plus on observe la forme de la fenêtre temporelle le long de la transformée .

Puis nous avons regardé la réponse de la fonction à une exponentielle complexe à 1000 Hz : cette fois-ci on est bien localisé en fréquence et étalé en temps (cf. figure 3.2).

Ce signal nous a permis aussi de vérifier que la formule de la "gaborette" (fenêtre d'analyse translatée en temps et en fréquence) $g_{k,l} = M_{l\nu_0} T_{kb_0} g(t) = g(t - kb_0).e^{2i\pi l\nu_0 t}$ est bien celle utilisée. En effet il existe une autre façon de calculer la transformée de Gabor : cela depend de l'ordre d'application des opérateurs de translation et de modulation sur la fenêtre g. On peut alors prendre comme gaborette $\tilde{g}_{k,l} = T_{kb_0} M_{l\nu_0} g(t) = g(t - kb_0).e^{2i\pi l\nu_0 (t-kb_0)}$.

Si on calcule les deux transformées possibles pour $\epsilon_\lambda(t) = e^{2i\pi\lambda t}$ on obtient [2] :

$$\begin{aligned} \text{avec } g_{k,l} : \langle s(t), g_{k,l} \rangle &= \int e^{2i\pi(l\nu_0 - \lambda)} t\overline{g}(t - kb_0)dt \\ &= e^{2i\pi(l\nu_0 - \lambda)kb_0} \overline{\hat{g}(l\nu_0 - \lambda)} \end{aligned}$$

$$\begin{aligned} \text{et avec } \tilde{g}_{k,l} : \langle s(t), \tilde{g}_{k,l} \rangle &= \int e^{2i\pi(l\nu_0 - \lambda)t} \overline{\tilde{g}}(t - kb_0)dt \\ &= e^{2i\pi(\lambda)kb_0} \overline{\hat{\tilde{g}}(l\nu_0 - \lambda)} \end{aligned}$$

Pour determiner quelle est la gaborette utilisée, il faut donc observer l'evolution de la phase de la transformée au cours du temps et à une fréquence donnée (la

[1] On peut faire un rapprochement entre M et ν_0 par $M = \frac{F_s}{\nu_0}$, F_s étant la fréquence d'échantillonnage du signal

[2] on remarquera que si $g_{k,l}$ est la gaborette utilisée pour le calcul de la transformée de Gabor d'un signal s(t), alors $v_g(s)(k,l) = \left\langle s, g_{k,l} \right\rangle$

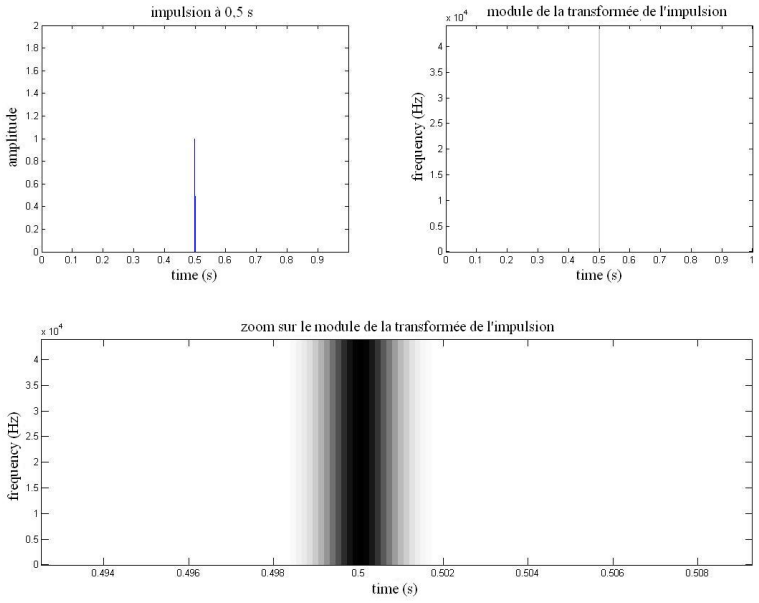

FIG. 3.1 – résultats obtenus avec une impulsion à $t_0 = 0,5$ s : sur le zoom on voit que l'on observe la forme de la fenêtre à travers le signal

fenêtre g étant réelle). En effet cette phase évolue linéairement[3] avec une pente de $2\pi(\lambda - l\nu_0)$ pour $g_{k,l}$ et avec une pente de $2\pi\lambda$ (pente qui ne dépend pas de $l\nu_0$) pour $g_{\tilde{k},l}$. Avec le signal précédent la phase evolue clairement en $2\pi(\lambda - l\nu_0)$ (cf. figure 3.3) et on a donc la confirmation que la transformée est bien calculée avec $g_{k,l}$.

Enfin, les derniers tests ont été effectués avec des signaux calibrés dont on

[3]sous $Matlab$© la phase est calculée entre $-\pi$ et π à l'aide de la fonction $atan2.m$. Afin de mieux voir son évolution on utilise alors la fonction unwrap.m qui permet de dérouler cette phase.

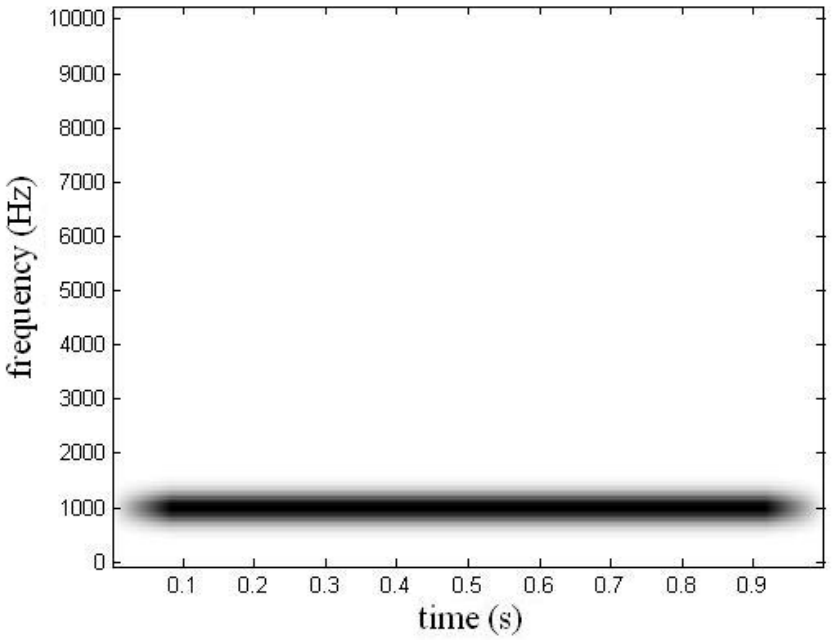

FIG. 3.2 – module de la transformée de Gabor d'une exponentielle complexe de fréquence 1000 Hz avec une attaque linéaire : l'energie est localisée autour de 1000 Hz sur toute la durée du signal.

a fait varier le temps d'attaque ou encore le rapport $\frac{\tau}{T}$ (temps d'attaque sur période du signal). Nous avons aussi vérifié à partir de quel seuil une variation de fréquence était détectable. Pour cela il a fallu coder une fonction test.m qui récupère les noms des fichiers, génère les signaux demandés, calcule et trace la transformée de Gabor de chaque signal. Tous ces résultats sont donnés dans les annexes au paragraphe : tests sur la toolbox ltfat.

Fonctions de génération des fenêtres de synthèse et d'analyse

Ici nous allons voir les différentes fonctions de la toolbox permettant de calculer les fenêtres d'analyse :

- la fonction pgauss.m ($g = pgauss(L, tfr)$) permet de générer une fenêtre gaussienne symétrisée (figure 3.4) de longueur L et ayant un rapport entre les supports temporel et fréquentiel de tfr[4]. L'écart type de g est donc calculé en fonction de la valeur de tfr. En effet si $tfr > 1$ alors g aura un support plus grand que sa transformée de fourrier discrete et à l'inverse si $tfr < 1$ c'est la transformée de g qui aura le plus grand support.

[4] on appel supports temporel et fréquentiel les longueurs effectives des fenêtres respectivement en temps et en fréquence

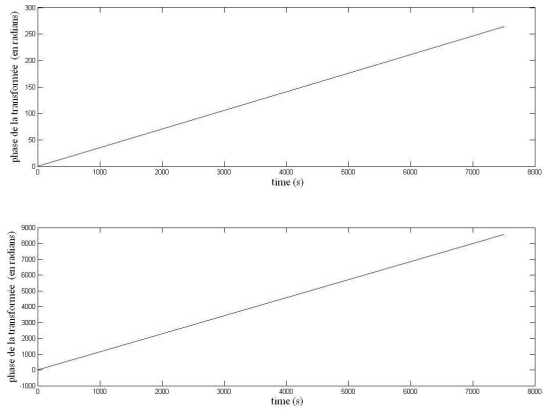

FIG. 3.3 – évolution de la phase au cours du temps pour $l\nu_0 = 970,2$ Hz en haut et $l\nu_0 = 0$ Hz en bas : les pentes sont positives de valeurs respectives $2\pi(\lambda - l\nu_0) \approx 186,2$ radians/s et $2\pi\lambda \approx 6283$ radians/s

- la fonction candual.m ($gamma = candual(g, a, M)$) permet de calculer la fenêtre duale canonique de g, c'est à dire la fenêtre de synthèse de la transformation inverse de Gabor. On voit que cette fenêtre dépend de la longueur du maillage en temps et en fréquence (b_0 et ν_0 correspondant ici à a et $\frac{F_s}{M}$).
- la fonction cantight.m ($gamma = candual(g, a, M)$) est utile lorsqu'on souhaite se placer dans un repère strict et donc avoir $g = h$. Elle permet de trouver à partir d'une fenêtre g calculée auparavant, la fenêtre la plus proche de g permettant d'obtenir ces conditions.

De plus la fenêtre d'analyse doit être choisie avec précaution en fonction de ce que l'on veut faire resortir du signal. En effet si on choisit mal cette fenêtre, il se peut que l'on n'observe plus le signal à travers la fenêtre mais le contraire. Par exemple le support de g en temps et en fréquence agit clairement sur la précision dans les deux dimensions habituelles : Plus le support de la fenêtre temporelle sera étalée et moins la transformée d'une impulsion sera localisée en temps. De même, moins le support fenêtre sera étalée en temps et moins la transformée d'un signal stationnaire sera localisée en fréquence (cf. figure 3.5).

Le meilleur compromis est de prendre $tfr = \frac{b_0 M}{L}$ où b_0 est la longueur de grille en temps, M le nombre de modulations et L la longueur de la fenêtre [5].

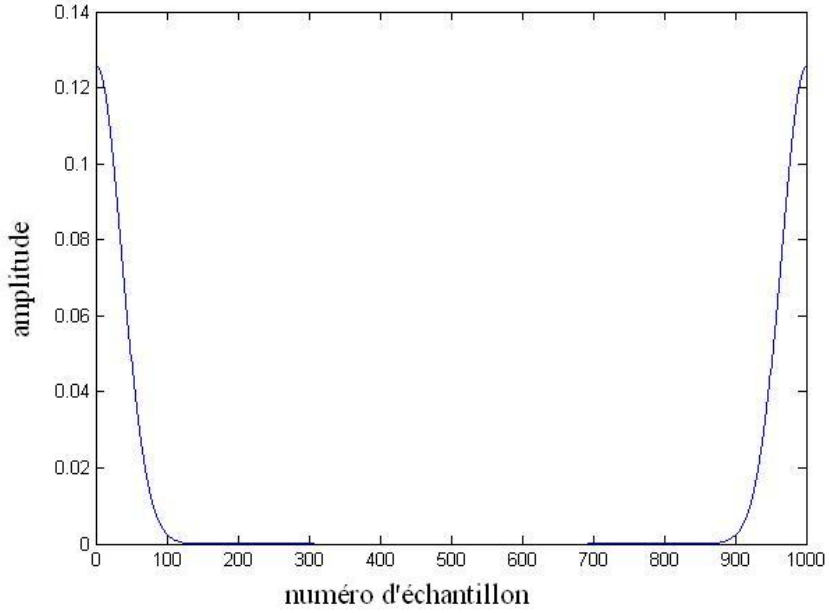

FIG. 3.4 – fenêtre gaussienne symétrisée d'une longueur de 1000 échantillons et ayant un rapport $tfr = 8$

3.3 Les multiplicateurs de Gabor

3.3.1 Définition

Etant donné une suite (à deux indices) $(k, l) \rightarrow \underline{m}(k, l)$, et les fenêtres g et h, on définit le multiplicateur $M_{g,h,\underline{m}}$ par

$$M_{g,h,\underline{m}}(s) = \sum_{k,l} m(k,l) v_g(s)(k,l) h_{k,l} \tag{3.3}$$

Par définition $\underline{\text{m}}$ est appelé masque de Gabor.C'est une matrice ayant les mêmes dimensions que la transformée de Gabor du signal.
On effectue donc la transformée de Gabor inverse de $v_g(s)$ multipliée terme à terme avec le masque.
On remarque alors que le multiplicateur $M_{g,h,\underline{m}}$ est un opérateur dans l'espace signal alors que le masque $\underline{\text{m}}$ est l'opérateur linéaire correspondant dans l'espace des transformées de Gabor.

3.3.2 Utilisation

Une des applications première et la raison d'être des multiplicateurs de Gabor est de trouver la meilleure approximation d'un signal s_1 par un multipli-

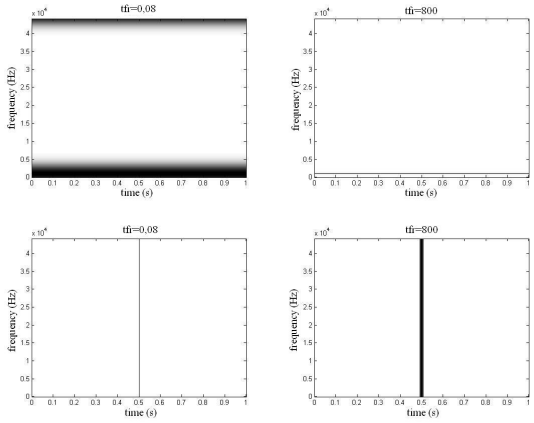

FIG. 3.5 – Les deux courbes du haut montre les modules des transformées de Gabor d'un signal stationnaire avec une fenêtre ayant un $tfr = 0,08$ (c'est à dire ayant un support en fréquence $12,5$ fois plus grand que son support temporel) à gauche et un $tfr = 800$ (c'est à dire ayant un support temporel 800 fois plus grand que le support fréquentiel) à droite. Les deux courbes du bas montrent les modules des transformées d'une impulsion à $t_0 = 0,5$ s pour les mêmes valeurs de tfr

cateur appliqué à un signal de référence s_0. D'autres applications sont envisageables comme par exemple le traitement d'un signal par un autre ou encore l'extraction d'un paramètre invariant pour une certaine classe de signaux.
Dans le cadre de cette étude on se place dans un repère strict : on a alors l'égalité entre la fenêtre d'analyse et la fenêtre de synthèse.
Il reste ensuite à identifier les masques correspondant a une transformation donnée. Les calculs de ces masques nous serviront alors à identifier les transitoires et à modéliser nos signaux par une succession de transformations sur un signal de référence.

Chapitre 4

Optimisation des masques de Gabor pour caractériser les transitoires

4.1 Intérêt pour caractériser les transitoires

L'intérêt de travailler avec les masques de Gabor dans l'analyse des transitoires est d'en avoir une nouvelle vision : celle d'un signal auquel on a appliqué des transformations grâce au masque de Gabor ainsi que des transpositions en temps et en fréquence.
Afin de calculer ces masques il est nécessaire de procéder par optimisation car on ne peut pas en connaître la valeure exacte en raison de régions temps-fréquence potentiellement d'énergie nulle. Pour optimiser ces différents paramètres, on travaillera avec la méthode des moindres carrés afin de minimiser les différences d'energie.

4.2 Calcul du masque sans recalage

On suppose, tout d'abord, que l'on applique aucune modulation ni aucune translation au signal avant d'estimer le multiplicateur. On cherche alors a minimiser la distance entre le signal cible s_1 et le signal de référence s_0 auquel on aura appliqué le multiplicateur :

$$min_{\underline{m}} \left(\left\| s_1 - M_{g,h,\underline{m}}(s_0) \right\|^2 \right)$$

en passant dans l'espace des transformées on obtient cette nouvelle expression :

$$min_{\underline{m}} \left(\left\| v_g(s_1) - \underline{m} v_g(s_0) \right\|^2 \right)$$

La méthode pouvant donner des valeurs de masques très importantes dans les zones ou le signal de départ s_0 possède peu d'énergie, on la stabilise en introduisant une contrainte sur la norme du masque :

$$min_{\underline{m}} \left(\|v_g(s_1) - \underline{m} v_g(s_0)\|^2 + \lambda \|m\|^2 \right)$$

où λ est le paramètre de Lagrange. Soit :

$$min_{\underline{m}(k,l)} \left(\sum_{k,l} \left\{ |v_g(s_1)(k,l) - \underline{m}(k,l) v_g(s_0)(k,l)|^2 + \lambda |\underline{m}(k,l)|^2 \right\} \right)$$

En dérivant cette expression par rapport à chaque composante $\underline{m}$(k,l) du masque et en annulant cette dérivée pour trouver le minimum on obtient :

$$\frac{\partial \left(\sum_{k,l} \left\{ |v_g(s_1)(k,l) - \underline{m}(k,l) v_g(s_0)(k,l)|^2 + \lambda |\underline{m}(k,l)|^2 \right\} \right)}{\partial(\underline{m}(k,l))} = 0$$

$$\Rightarrow -2\overline{v_g(s_0)}(k,l) \left[v_g(s_1)(k,l) - \underline{m}(k,l) \right] + 2\lambda \underline{m}(k,l) = 0$$

$$\Rightarrow \underline{m}(k,l) \left[|v_g(s_0)(k,l)|^2 + \lambda \right] = \overline{v_g(s_0)}(k,l) v_g(s_1)(k,l)$$

$$\Rightarrow \boxed{\underline{m}(k,l) = \frac{\overline{v_g(s_0)}(k,l) v_g(s_1)(k,l)}{|v_g(s_0)(k,l)|^2 + \lambda}} \qquad (4.1)$$

On voit alors que le masque n'est autre qu'une fonction de transfert entre les deux transformées. De plus on remarque que lorsque le module de $v_g(s_1)$ est très petit (devant λ) alors c'est λ qui prend le dessus au dénominateur et empêche ainsi $\underline{m}$ de diverger. Mais le problème est que si les signaux ne sont pas recalés entre eux dans le domaine temps-fréquence le masque ne pourra pas créer de l'énergie là où il n'y en a pas et par conséquent l'approximation sera très mauvaise (cf. figures **??**).

4.3 Calcul du masque avec recalage

On suppose maintenant que pour recaler le signal s_0 (signal de référence) avec le signal s_1 (signal cible), on doit lui appliquer un opérateur $T_{k',l'}$ qui réalise une translation en temps de $k'b_0$ et une modulation de $l'\nu_0$ telles que la différence d'énergie entre s_1 et s_0 recalé soit la plus petite possible. On cherche alors le multiplicateur $M_{g,h,\underline{m}}$ tel que :

$$s_1 = M_{g,h,\underline{m}} T_{k',l'}(s_0)$$

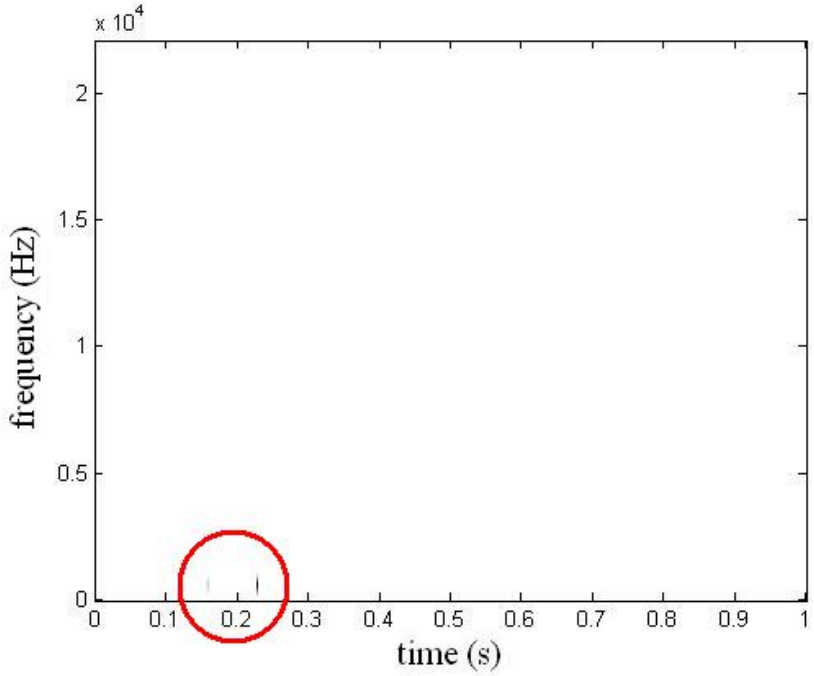

FIG. 4.1 – module du masque entre deux signaux non recalés : le masque ne peut pas reconstruire le signal de façon adéquate car il n'y a pas d'energie là ou il est calculé : On a alors un masque nul partout sauf dans les basses fréquences autour de $0,2$ s (On remarquera que le masque n'est calculé que pour les fréquences de 0 à $\frac{F_s}{2}$, F_s étant la fréquence d'échantillonnage du signal, ceci pour alléger les calculs et éviter les problèmes de mémoire).

En appliquant la même méthode que précédemment, on trouve l'expression à minimiser :

$$min_{\underline{m},k',l'}\left(\sum_{k,l}\left\{|v_g(s_1)(k,l)-\underline{m}(k,l)T_{k',l'}(v_g(s_0))(k,l)|^2+\lambda\,|\underline{m}(k,l)|^2\right\}\right)$$

En supposant que l'on connait k' et l', on obtient le masque :

$$\boxed{\underline{m}(k,l)=\frac{\overline{v_g(T_{k',l'}(s_0))}(k,l)v_g(s_1)(k,l)}{|v_g(T_{k',l'}(s_0))(k,l)|^2+\lambda}}$$

Or $vg(T_{k',l'}(s_0))$ peut s'exprimer sous forme d'une intégrale ce qui permet de continuer le calcul :

$$\langle T_{k',l'}(s_0),g_{k,l}\rangle=\int s_0(t-k'b_0)e^{2i\pi l'\nu_0 t}\overline{g(t-kb_0)e^{2i\pi l\nu_0 t}}dt$$

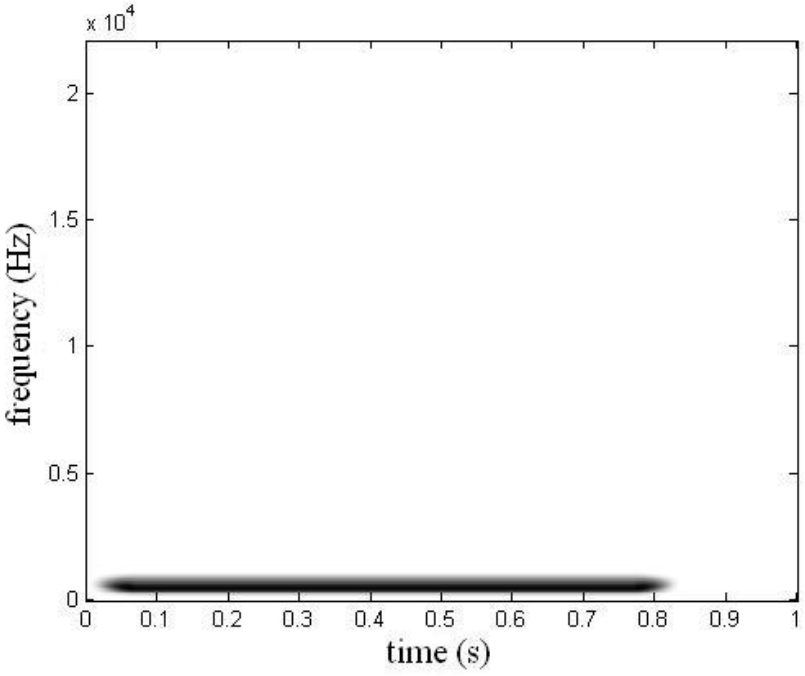

FIG. 4.2 – module du masque entre deux signaux recalés : le masque reconstruit bien le signal cible. Ici on a pris pour signal de départ une sinusoïde avec une attaque linéaire et comme signal cible une autre sinusoïde avec une attaque en cosinus. Le masque va alors diminuer l'énergie du signal aux extrémités ce qui explique pourquoi ses côtés ont une valeur plus petite que 1.

$$= \int s_0(t - k'b_0)\overline{g}(t - kb_0)e^{2i\pi(l'-l)\nu_0 t}dt$$

on effectue le changement de variable : $t_1 = t - k'b_0$

$$= \int s_0(t_1)\overline{g}(t_1 - (k - k')b_0)e^{2i\pi(l'-l)\nu_0(t_1+k'b_0)}dt_1$$

$$= e^{2i\pi(l'-l)\nu_0 k'b_0} \int s_0(t_1)\overline{g}(t_1 - (k - k')b_0)e^{2i\pi(l'-l)\nu_0 t_1}dt_1$$

$$\boxed{\langle T_{k',l'}(s_0), g_{k,l}\rangle = v_g(s_0)(k - k', l - l').e^{2i\pi(l'-l)\nu_0 k'b_0}}$$

Ceci nous permet d'exprimer $\underline{m}(k,l)$ en fonction de $v_g(s_0)$:

$$\boxed{\underline{m}(k,l) = \frac{\left(e^{2i\pi(l-l')\nu_0 k' b_0}\overline{v_g(s_0)}(k-k',l-l')\right) v_g(s_1)(k,l)}{|v_g(s_0)(k-k',l-l')|^2 + \lambda}} \quad (4.2)$$

Par rapport au résultat obtenu sans le recalage, on remarque l'apparition d'un terme correctif de phase ($e^{2i\pi(l-l')\nu_0 k' b_0}$) permettant d'éviter les interférences entre $v_g(s_0)(k-k',l-l')$ et $v_g(s_1)(k,l)$. Ce terme doit être appliqué à $v_g(s_0)$ avant le calcul du masque.

4.4 Les différentes méthodes de recalage

Afin de calculer k' et l' nous avons utilisé plusieurs méthodes qui vont être présentée ci-dessous. Ces différentes méthodes ont chacune leurs avantages et leurs inconvénients. Elles diffèrent essentiellement par la technique d'identification des paramètres de décalages en temps et en fréquence optimaux.

4.4.1 Recalage par centre de gravité

Cette première méthode consiste à calculer le centre de gravité des transformées des deux signaux et à décaler la transformée de s_0 afin que son centre coïncide avec celui de la transformée de s_1 (cf. figure 4.3). Si (i_0, j_0) et (i_1, j_1) sont les indices respectifs des centres de gravité de $v_g(s_0)$ et $v_g(s_1)$ alors on peut calculer k' et l' avec :

$$k' = i_1 - i_0$$
$$l' = j_1 - j_0$$

Le problème de la méthode est qu'elle ne fonctionne pas bien pour des signaux dont l'énergie est localisé à plusieurs endroits dans le plan temps-fréquence (cf. figure 4.4). En effet, dans ce cas le centre de gravité du signal peut se trouver à un endroit où il y a peu d'énergie. On aura alors le même problème que si les signaux n'étaient pas recalés.

4.4.2 recalage par corrélation

utilisation de la corrélation croisée

Pour obtenir un résultat plus adapté, on peut procéder par optimisation en essayant de minimiser $\|v_g(s_1) - v_g(T_{k',l'}(s_0))\|^2$. Cela revient à chercher le maximum de $\langle v_g(s_1), v_g(T_{k',l'}(s_0))\rangle$ c'est a dire :

$$max_{k',l'}\left(\sum_{k,l} v_g(s_1)(k,l)\underline{v_g}(s_0)(k-k',l-l')e^{-2i\pi(l-l')\nu_0 k' b_0}\right)$$

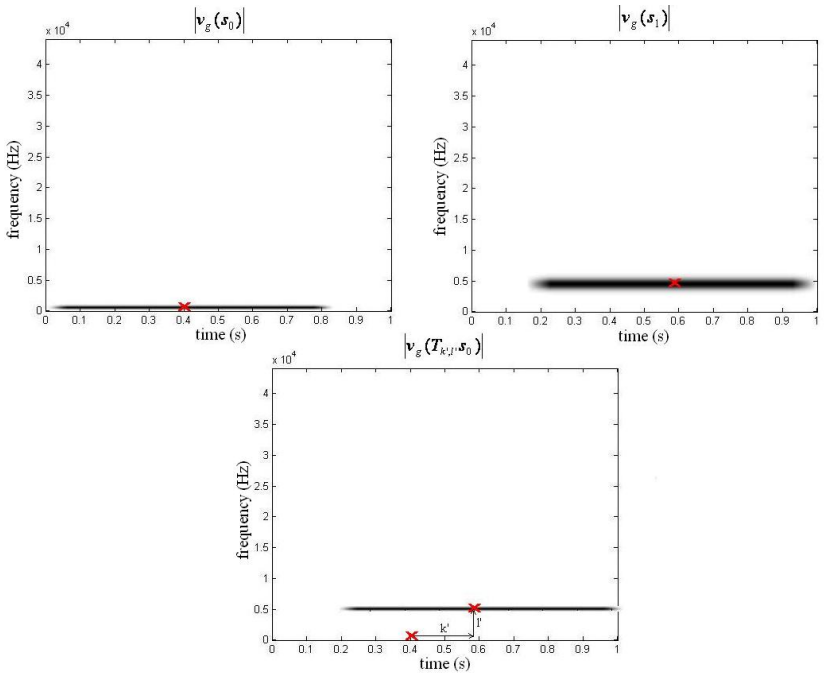

FIG. 4.3 – exemple de recalage par centre de gravité : on à pris ici un signal de départ à 440 Hz sans retard et un signal cible à 4400 Hz retardé de 0,2 seconde. Les centres de gravité de chaque transformée sont marqués d'une croix.

L'expression entre parenthèses correspond au calcul de la fonction xcorr2.m ($C = xcorr2(A, B)$) sous $Matlab^{©}$ au terme exponentiel près. En effet c'est une corrélation croisée entre les deux transformée. Donc si on travaille avec les modules, il suffit d'utiliser cette fonction et de trouver les indices de la valeur maximum de la matrice de sortie pour avoir k' et l'.
Cette méthode donne de bons résultats même pour les signaux de la figure 4.4, mais elle est beaucoup trop coûteuse en temps de calcul car les matrices des transformées sous $Matlab^{©}$ ont des dimensions trop importantes. Une solution possible est de recaler par centre de gravité puis de calculer une corrélation limitée aux termes centraux afin d'affiner le résultat. Mais dans ce cas, on peut parfois observer le même problème que celui évoqué au paragraphe précédent.
Une autre alternative consiste à appliquer un filtre passe bas sur les deux transformées puis de les sous-échantillonner avant de calculer leur corrélation. On n'est alors pas aussi précis qu'en calculant la corrélation entière mais les endroits où l'énergie est localisée sont quand même recalés correctement entre les deux transformées. Il faut alors trouver le bon compromis pour avoir un temps de calcul minimal sans trop altérer les résultats : il semble qu'un sous-échantillonnage avec un pas de 10 est raisonnable (cf. figure 4.5).

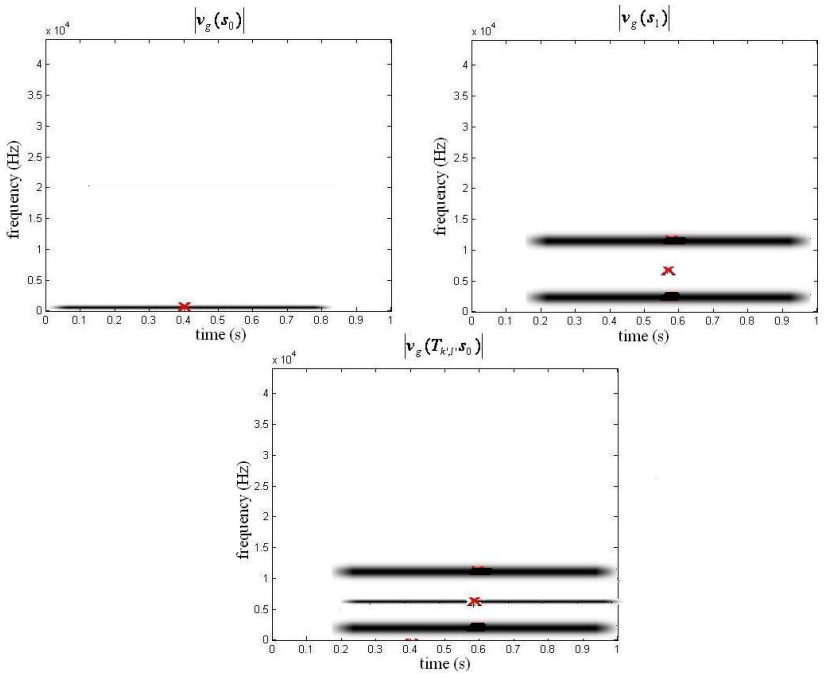

FIG. 4.4 – On est ici dans le cas où les deux transformées recalées entre elles n'ont pas d'énergie en commun. c'est dans ce cas que le masque peut diverger si on ne lui applique pas λ et qu'une autre méthode de recalage est nécessaire.

Utilisation de la transformée de Gabor

Si on revient au produit scalaire $\langle v_g(s_1), v_g(T_{k',l'}(s_0))\rangle$ on voit que d'après la conservation de l'énergie par transformée, chercher son maximum revient à chercher le maximum en (k', l') de $\langle s_1, T_{k',l'}s_0\rangle$. Or ce produit scalaire n'est autre que la transformée de Gabor du signal s_1 en utilisant comme fenêtre d'analyse le signal s_0. Donc pour trouver k' et l' il ne reste plus qu'à trouver les coordonnées du maximum. Ce calcul est beaucoup plus rapide que le calcul de corrélation croisée et il reste précis. Il faut par contre faire attention aux valeurs de b_0 et ν_0 que l'on à choisies afin de trouver les bonnes valeurs des décalages.

4.5 Algorithme itératif

On a vu que lorsque la transformée du signal s_0 et la transformée du signal s_1 n'avaient pas leur énergie au même endroit dans le plan temps-fréquence, le masque n'arrivait pas à reconstruire correctement le signal s_1. Ceci est valable notamment lorsque $v_g(s_0)$ est moins étalée que $v_g(s_1)$ (par exemple si s_0 possède une seule composante harmonique et que s_1 en a plusieurs). Dans ce cas le

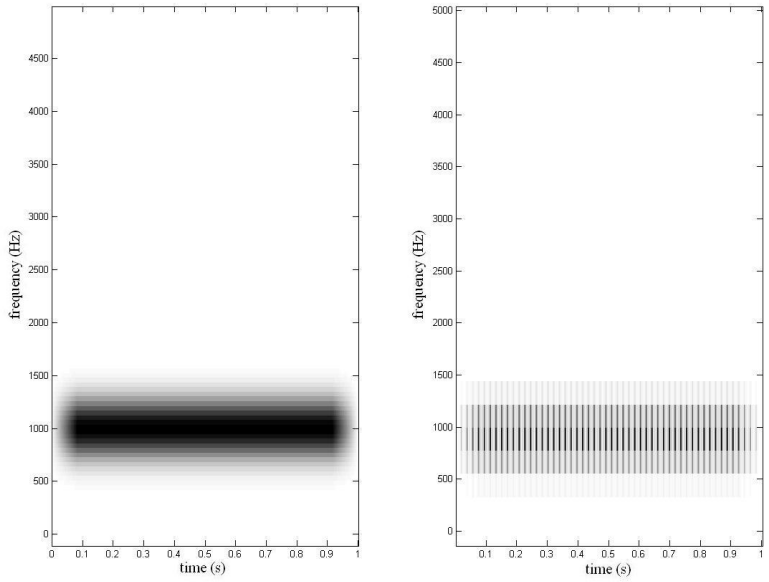

FIG. 4.5 – La figure de gauche représente la transformée de départ et celle de droite la transformée filtrée passe-bas puis sous-échantillonnée avec un pas de 10 : on ne fait alors plus attention au détails (hautes fréquences) lors du calcul de la corrélation.

recalage ne suffira pas et s_1 ne sera reconstruit que partiellement (dans l'exemple précédent on ne reconstruira alors qu'une seule harmonique).
Nous avons donc mis en place un algorithme qui consiste à soustraire le signal reconstruit au signal cible s_1 afin d'obtenir un résidu r_1 puis de calculer un nouveau masque pour passer de s_0 à r_1 ainsi que des nouveaux décalages. Le signal s_1 est alors vu comme une combinaison linéaire de masques appliqués à s_0 recalé. On peut répéter le processus qui mène à l'algorithme itératif suivant :

forme de l'algorithme :

Si on note $\underline{m} = masque(a, b)$ la fonction de calcul du masque pour passer de la transformée du signal a à la transformée du signal b alors on obtient :
Pour p *de* 1 *à* N_{iter} *faire*

$$\begin{aligned} \underline{m}_p &= masque(s_0, r_{p-1}) \\ s_{r,p} &= M_{\underline{m}_p,g}(T_{k'_p,l'_p}(s_0)) \\ r_p &= r_{p-1} - s_{r,p} \end{aligned}$$

Finpour

$$sr = \sum_p \left(M_{\underline{m}_p, g} T_{k'_p, l'_p} \right) (s_0)$$

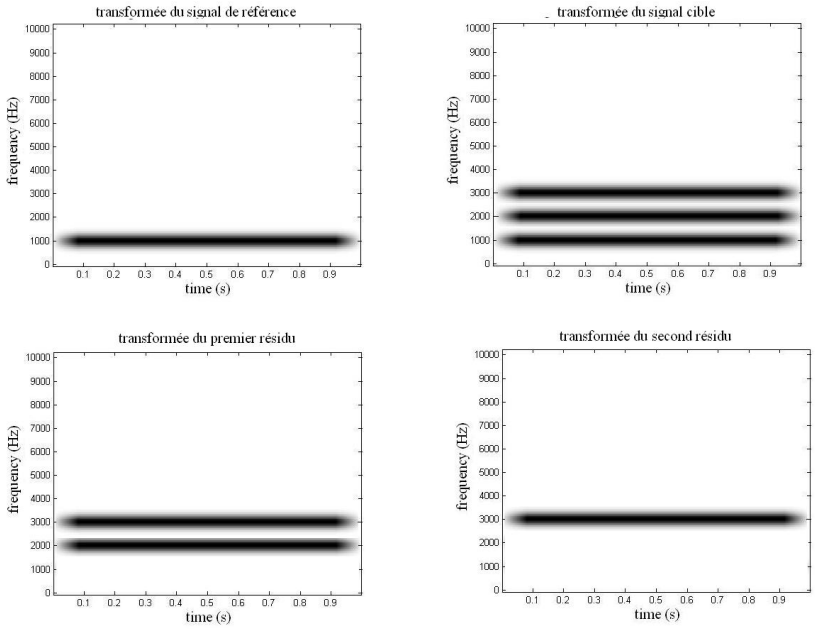

FIG. 4.6 – Application de l'algorithme itératif pour retrouver un signal à trois harmoniques à partir d'un signal n'en possédant qu'une seule.

Conclusion

Les formes de transitions simples ont permis de tester et valider la méthode de recalage par centre de gravité. Nous parvenons donc maintenant à exprimer un transitoire en fonction d'une somme de multiplicateurs appliqués à un signal de référence. Cela nous permet d'observer les transformations à appliquer pour passer d'un transitoire à un autre et ainsi de les caractériser.
Mais la méthode s'avère plus générale et elle pourra permettre de caractériser des types de sons bien définis. En effet lors de mon stage j'ai aussi travailler en collaboration avec un autre étudiant dont le travail était de caractériser des sons liés à un mouvement.
Lors des deux prochains mois il faudra ensuite travailler sur des signaux réels en les comparants avec les signaux calibrés que nous avons puis entre eux. Il faudra aussi traiter le cas de l'estimation d'un masque à partir de plusieurs réalisations car cela nous permettra d'obtenir un seul masque pour des signaux de même type.
De plus nous avons traité seulement deux transformations dans le plan temps-fréquence (la modulation et la translation) mais d'autres transformations sont possible comme la dilatation ou encore la rotation. Bien que l'on ne puisse pas traiter ces transformations par les multiplicateurs de Gabor, elles n'en restent pas moins fondamentales en acoustique. En effet, si on prend l'exemple d'une note de musique, la représentation dans le plan temps-fréquence d'une autre note jouée sur le même instrument est à peu de chose près la même mais dilatée. Le formalisme des multiplicateurs en ondelettes paraît donc incontournable pour la suite.

Bibliographie

[1] K. Shnass. *Gabor multipliers, a self-contained survey.* rapport de Master. université de Vienne. 2004.

[2] T. Tolonen, V. Välimäki, and M. Karjalainen. *Evaluation of modern sound synthesis methods.* rapport 48. Helsinki university of Technology, Departement of electrical and communications engieering, Laboratory of acoustics and audio signal processing. Mars 1998.

[3] L. Daudet. *A review on techniques for the extraction of transients in musical signals.* rapport CMMR. université Pierre et Marie Curie (Paris 6), Laboratoire d'accoustique musicale. 2005.

[4] F.J. Harris. *On the use of windows for harmonic Analysis with the discrete Fourier transform.* Proceedings of the IEEE, Vol.66,NO.1. Janvier 1978.

[5] P.L. Sondergaard. *The linear time-frequency analysis toolbox.* rapport non classifié (www.univie.ac.at/nuhag-php/ltfat). université de Vienne. 2006.